Express Mail Label: EM049017355US
Date of Deposit: June 23, 1997

ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

CASE DOCKET NO. YO997-111
Date: June 23, 1997

Sir:

Transmitted herewith for filing is the Patent Application of:

Inventors: Mitchell A. Cohen, Anant D. Jhingran and Ronald Mraz

For: SYSTEM AND METHOD FOR MAXIMIZING USAGE OF COMPUTER RESOURCES IN SCHEDULING OF APPLICATION TASKS

Enclosed are:

Express Mail Label No.: EM049017355US
Date Deposited: June 23, 1997

_X_ _5_ Sheets of Informal Drawings.

_X_ An assignment of the invention to International Business Machines Corporation and the assignment recordation form cover sheet.

___ A certified copy of a ______________________ application.

_X_ Declaration and Power of Attorney.

The filing fee has been calculated as shown below:

FOR:	NO. FILED (Col. 1)	NO. EXTRA (Col. 2)
BASIC FEE		
TOTAL CLAIMS	18-20 =	0
INDEP CLAIMS	2- 3 =	0
____ MULTIPLE DEPENDENT CLAIM PRESENTED		

If the difference in Col. 1 is less than zero, enter "0" in Col. 2.

OTHER THAN A SMALL ENTITY

RATE	FEE
	$770.00
X $ 22 =	$ 0
X $ 80 =	$ 0
+ $ 240=	$ 0.00
SUBTOTAL	$ 770.00
	$
TOTAL	$ 770.00

_x_ Please charge my Deposit Account No. 09-0468 in the amount of $ 770.00 .

_x_ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 09-0468. A duplicate copy of this sheet is enclosed.

_x_ Any additional filing fees required under 37 CFR 1.16.

_x_ Any patent application processing fees under 35 CFR 1.17.

Respectfully submitted,

By [signature]

Attorney: Douglas W. Cameron
Registration No.: 31,596
Tel. (914) 945-3244

IBM CORPORATION
INTELLECTUAL PROPERTY LAW DEPT.
P.O. BOX 218
YORKTOWN HEIGHTS, NY 10598

Express Mail Label: EM049017355US
Date of Deposit: June 23, 1997

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Mitchell A. Cohen et al. Date: June 23, 1997

Serial No.: Group Art Unit:

Filed: Herewith Examiner:

For: SYSTEM AND METHOD FOR MAXIMIZING USAGE OF COMPUTER RESOURCES IN SCHEDULING OF APPLICATION TASKS

ASSOCIATE POWER OF ATTORNEY

The Commissioner of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

The undersigned attorney, who has been appointed as an attorney in the combined Declaration and Power of Attorney in the above-identified application, hereby appoints Anne V. Dougherty, Reg. No. 30,374, as attorney to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. All correspondence should be directed to Anne V. Dougherty, 3173 Cedar Road, Yorktown Heights, New York 10598. Phone calls should be directed to Anne Dougherty at (914) 962-5910.

Respectfully Submitted,

Douglas W. Cameron
Attorney for Applicants
Reg. No. 31,596
Tel. (914) 945-3244

IBM Thomas J. Watson Research Center
Intellectual Property Law Department
P.O. Box 218 - 39-254
Yorktown Heights, N.Y. 10598

Express Mail Label: EM049017355US
Date of Deposit: June 23, 1997

SYSTEM AND METHOD FOR MAXIMIZING USAGE OF COMPUTER RESOURCES IN SCHEDULING OF APPLICATION TASKS

Field of the Invention

This invention relates to scheduling of applications among processes in one or more associated computers; and, more particularly, to a scheduling system which implements a task schedule by setting operating system priorities for the processes working on queued activities to optimize usage of shared computer resources.

Background of the Invention

Scheduling of activities is needed when a computer is running multiple activities or applications. Assuming that each application or activity comprises more than one task, the tasks must be scheduled among available processes of the computer, often with the order of tasks being predetermined based upon the requirements of the application. For example, when doing a merge-sort operation, the tasks of sorting records into lists must be performed before the next task of merging the lists. The scheduling task becomes more challenging in a multi-processor

parallel computing environment, where multiple tasks may be run simultaneously by associated processes. For optimal usage of the available resources, processes should have waiting tasks queued for commencement as soon as previous tasks have been completed, with "wait states" being filled in with queued tasks.

In the past, load control has been used for multi-process scheduling. Under a load control scheduling scheme, only a subset of the total number of tasks are allowed to run at one time. If the processes for each of the subset of tasks all enter wait states (for example, pending the completion of a parallel-running task of the application by another process), the CPU will be unused throughout the duration of the wait states, even though there is more work queued. Scheduling of too few activities under the load control mechanism, therefore, frequently leads to underutilization of the CPU. On the other hand, if too many activities are allowed to run at once under a load control scheme, which is done under the assumption that all activities will not enter wait states at the same time, the ability to schedule among all of the tasks which are running is lost.

Another scheduling method which has been used in the prior art is priority-based scheduling for management of computer resources. Under a priority-based scheduling scheme, an operating system scheduler prioritizes the workload and schedules one task to be active at any given time. For example, on the AIX* (* Trademark of International Business Machines Corporation) operating system, if applications A, B and C are to be scheduled in alphabetical order,

and processes 1, 3 and 5 are working on A, 2 and 4 on B, and 6 on C, then processes 1, 3 and 5 have their operating system priorities set to 60, 2 and 4 to 61, and 6 to 62 (where lower process priority is more favorable). The prioritization scheme is effectively a resource utilization mechanism that does not perform scheduling of prioritized tasks among processes with the intent of running one or more applications as quickly and efficiently as possible.

When an activity to be scheduled does not parallelize into even amounts of work, neither load nor uniprocessor priority scheduling can maximize the application throughput. Database management systems, wherein the amount of work necessary for any task cannot be quantified in advance without detailed knowledge of the database and of the transactions to be performed thereon, defy scheduling by load or uniprocessor prioritization. Ideally, the scheduler must provide the ability to continue on to other activities related to the initial task when part of a parallel activity has been completed yet other related parts have not been completed.

What is desirable, therefore, is a dynamic priority scheduling mechanism for scheduling activities with multiple tasks among multiple processes, for minimizing unused CPU time.

It is therefore an objective of the present invention to implement scheduling at the task level.

It is additionally an objective of the present invention to provide multiple task scheduling which will minimize unused CPU time.

Still another objective of the invention is to provide coordination of activities among parallel computational resources to minimize unused CPU and optimize application run-time.

Summary of the Invention

These and other objectives are realized by the present invention wherein a task schedule is enforced among multiple processes by setting process priorities based upon which tasks are running on which processes and based upon the task schedule. The task scheduling may be provided by a local or global scheduler which uses application information to prioritize tasks. The task schedule, or priority list, is provided at Local Activity Schedulers which schedule the activities for their local execution elements/nodes. Execution of activities locally are performed by any number of processes that reside in each execution element. These processes are assigned operating system priorities by the respective Local Activity Scheduler based on their assigned activities for execution and the task schedule.

Brief Description of the Drawings

The invention will now be described in detail with specific reference to the attached drawings wherein:

Figure 1 provides a schematic illustration of a parallel processing system utilizing one embodiment of the present invention.

Figure 2 illustrates the Activity Priority list maintained in accordance with the present invention.

Figure 3 illustrates the Local Activity Scheduler list maintained by the Global Activity Scheduler of the present invention.

Figure 4 illustrates the Activity-Process Correspondence Table maintained in accordance with the present invention.

Figure 5 provides a flow chart representative of the operations of the Global Activity Scheduler of the present invention.

Figure 6 provides a flow chart representative of the operations of the Local Activity Schedulers of the present invention.

Figure 7 provides a Gantt Chart of prior art scheduling of 4-way parallel activities.

Figure 8 provides a Gantt Chart of scheduling of 4-way parallel activities in accordance with the present invention.

Description of the Preferred Embodiment

One embodiment of the inventive multiple activity scheduling system for a parallel processing environment is illustrated in

Figure 1. As shown therein, a Global Activity Scheduler 10, utilizing information received from the applications as provided via the Application Coordinator 11, provides a prioritized schedule of tasks or activities along communication links to nodes 16-19. Each node is provided with a Local Activity Scheduler 12-15 which schedules each of the its associated processes, 102-109. The schedule information may be regularly updated based upon incoming activities to be scheduled and based upon information provided by continual monitoring of the resources at the nodes. As illustrated, communications between the entities are bi-directional, with the Local Activity Schedulers continuously reporting process information to the Global Activity Scheduler directly or via updates through the Application Coordinator, as will be further detailed below. While the system has been illustrated to include four nodes, each having two processes, it will be apparent that the present invention can be applied to a system having any number of nodes in communication with a Global Activity Scheduler, wherein each node may have any number of associated processes. Each node in the system is necessarily provided with a dedicated Local Activity Scheduler, which may or may not be physically located at the node. If the Local Activity Scheduler which is associated with a particular node is not physically located at that node, it is understood that the Local Activity Scheduler would be in constant communication with the operating system at the node. The Local Activity Scheduler is responsible for establishing the operating system priorities for

implementing the task schedule at the node. In an alternative embodiment, the Local Activity Scheduler may itself establish the task schedule, if no global entity is available or required, as further detailed below.

In the illustrated embodiment, for each activity to be scheduled, an activity ID is created. The activity ID can be the application command string, the user ID running the command, or an ID created by the application or by some other process at the Application Coordinator or Global Activity Scheduler. In addition, each process in a node has a process ID. When a process begins or ends its work on a task, it reports its activity ID and process ID to the Local Activity Scheduler process, which in turn reports the activity ID directly to the Global Activity Scheduler in a Begin-End Task message or indirectly via future updates from the Application Coordinator. The Global Activity Scheduler is provided with knowledge of which applications are in the system. Then the Global Activity Scheduler creates a schedule based on a scheduling algorithm (or uses a pre-determined schedule) for the application tasks and forwards this schedule to each Local Activity Scheduler associated with each node of the parallel computer.

The Local Activity Schedulers are each responsible for tracking which processes are working on which applications at their node. Using this knowledge and the task schedule (hereinafter referred to as the Activity Priority list), each Local Activity Scheduler determines the priorities for each of the processes on its node, and directs the local operating system to set the process

priorities for optimal execution of the prioritized activities.

All of the processes working on tasks for the same activity will get the same priority, with each activity being assigned a unique priority. When no more unique priorities exist, the activities scheduled at the end can all be given the least favored priority. When all the processes in the highest prioritized activities are not using CPU, the operating system will have the activity with the second highest priority use the CPU, and so on, to thereby limit the amount of unused CPU time.

For the Figure 1 embodiment, Figures 2 and 3 provide representative examples of two internal structures which would be dynamically maintained at the Global Activity Scheduler, specifically the Activity Priority list and the Local Activity Scheduler list. The Local Activity Scheduler list of Figure 3 comprises a list of all active Local Activity Schedulers in the system which are under the control of the Global Activity Scheduler (for example, at all nodes of a partition). As shown in Figure 3, the list includes the Local Scheduler ID along with its location address. The Local Activity Scheduler list is maintained for utilization when broadcasting priority information and is continually updated by communications received from the Local Activity Schedulers.

The Activity Priority list of Figure 2 is the task schedule which is derived from communication with the applications (from the Application Coordinator in the Figure 1 implementation), as to which activities are active in the system. The Activity Priority

list provides the activity IDs in priority order. Reception of Begin and End application messages from the Application Coordinator allows the scheduling program at the Global Activity Scheduler, or at the Local Activity Scheduler in the alternative embodiment, to maintain the Activity Priority list. The priorities of activities on the list can be determined by utilizing any number of parameters related to the activities to be scheduled. Examples of relevant parameters include the relative importance of the user activities (as provided by the user or a programmer), time deadlines, user expense guidelines, resource requirements, etc. Any message reception from the Application Coordinator triggers the relevant scheduling program out of its wait state to generate an updated Activity Priority list. In addition, completion of an activity, as communicated from the processes at a node, will cause removal of the activity from the Activity Priority list, and updating of the schedule. In the Figure 1 embodiment, the Global Activity Scheduler broadcasts the Activity Priority list to all Local Activity Schedulers either immediately upon generation of a new list or at periodic intervals.

At the Local Activity Scheduler, the latest version of the Activity Priority list is maintained, as communicated from the Global Activity Scheduler in the Figure 1 embodiment, or as derived locally in the alternative embodiment. In addition, the Local Activity Scheduler maintains an Activity-Process Correspondence table, as shown in Figure 4. The Activity-Process Correspondence table reflects the assignment of activities at the node to the

node's processes, along with the respective priorities of the activities. This information may be obtained directly from the processes themselves, under a task registration protocol, or indirectly, for example from the database monitor of a database on which processes are performing tasks. In the Figure 4 illustration, assuming a node having five available processes, the Activity IDs are paired with process IDs, with the respective activity priority assignments listed with the activity-process pairings. The priority assignments found on the Activity-Process Correspondence table are assigned by the Local Activity Scheduler based upon the Activity Priority list.

Figure 5 provides a representative process flow of the operations performed by the Global Activity Scheduler of the Figure 1 embodiment. At box 50, a communication packet is received at the Global Activity Scheduler, and, in step 51, the packet is analyzed to determine if the message is from the Local Activity Scheduler or from the Application Coordinator. If the message comprises a communication regarding processes from the Local Activity Scheduler, the Activity Scheduler list is updated to reflect the available processes, at step 52. If the message is from the Application Coordinator regarding completed tasks or tasks to be commenced, the scheduling program in invoked and the Activity Priority list updated at step 53. Depending upon the preferred programming order of operations, the Global Activity Scheduler may automatically send the updated Activity Priority list to all Local Activity Schedulers, as shown at step 54, or may wait until a pre-

set time interval has elapsed, as indicated by decision box 55, and then send the list. The Global Activity Scheduler then waits for the next communication packet, as shown at step 56.

Figure 6 illustrates the representative process flow of operations conducted by the Local Activity Scheduler of the Figure 1 embodiment. Upon receipt of a communication packet at 60, the Local Activity Scheduler determines, as indicated by decision box 61, whether the communication is from the Global Activity Scheduler or from one of its activity processes. If the message is from the Global Activity Scheduler, the message will contain an Activity Priority list to replace the previously-communicated list, as reflected at step 62. As noted above, the Local Activity Scheduler could, in an alternative embodiment, be the entity that establishes the task schedule, and would therefore automatically update its Activity Priority list. If the message is from an activity process, presumably either a Begin or End task message, then the Local Activity Scheduler updates the Activity-Process Correspondence table at step 63. In addition, at this juncture, the Local Activity Scheduler may communicate task-related messages (not shown) to the Global Activity Scheduler. Since communications between nodes and the applications/Application Coordinator regarding task commencement and completion are well known and need not be altered to implement the present invention, and since the Application Coordinator necessarily relays such information to the Global Activity Scheduler, it is not strictly necessary to incorporate the redundant step of the Local Activity Scheduler

communicating task commencement and completion messages to the Global Activity Scheduler.

Simultaneously with, or subsequent to, the appropriate updating of the Activity-Process Correspondence table, the Local Activity Scheduler assigns priorities, at step 64, and directs the local operating system to set the process priorities for execution of tasks at the node, at step 65. Finally, the Local Activity Scheduler enters a waiting state at step 66, awaiting receipt of the next communication packet.

By dynamically assigning priorities and allowing the local processes to move from one task to the next highest priority task without waiting, the system maximizes utilization of the CPU resources at each node. Figure 7 shows a Gantt Chart of Random scheduling of three 4-way parallel activities. In the prior art example, three parallel activities (A, B and C) are distributed across four processes or machines and are active at time zero. When each of the respective parallel parts are complete, the activity associated with the task is complete. The Random chart shows that each of the tasks is complete at time 30, and that the average time for completion is 30. This is because at least one parallel part of each activity was not able to begin until time 20. On the other hand, when using the inventive system, as illustrated on the Gantt chart of Figure 8, the average time to complete an activity would be 20, resulting in a speed-up of 33.3%, due to the parallel scheduling of activities among the processes.

It is to be noted that in a parallel environment wherein the

queued activities to be scheduled are similar in size and arrival time and are performing similar tasks, in terms of resource requirements, such as periodic display of fixed amounts of data retrieved from an outside source or database operations wherein the database is divided equally among the available processes, the activities of the Global Activity Scheduler could be performed by the Local Activity Schedulers, which would generate virtually identical schedules, thereby obviating the need for coordination of those schedules at the "global" level.

The invention has been described with reference to several specific embodiments. One having skill in the relevant art will recognize that modifications may be made without departing from the spirit and scope of the invention as set forth in the appended claims.

CLAIMS

Having thus described our invention, what we claim as new and desire to secure by Letters Patent is:

1. Apparatus for providing scheduling of a plurality of tasks of at least one application among processes in at least one computing node, each node having a plurality of local processes, comprising:

scheduler means for dynamically creating a prioritized schedule of said plurality of tasks; and

at least one local scheduler associated with said at least one computing node comprising means for ascertaining which of said plurality of tasks are assigned to each of said plurality of local processes and means for prioritizing said assigned processes in accordance with said prioritized schedule.

2. The apparatus of Claim 1 wherein said at least one computing node additionally comprises at least one operating system for receiving input from said means for prioritizing and for directing said assigned processes to execute said tasks in accordance with said prioritizing.

3. The apparatus of Claim 2 wherein said operating system is further adapted to interleave local operations with said tasks.

4. The apparatus of Claim 2 further comprising application coordinator means for communicating information about said plurality of tasks to said scheduler for use in dynamically creating said schedule.

5. The apparatus of Claim 2 wherein said local processes are adapted to perform tasks in parallel.

6. The apparatus of Claim 1 wherein said scheduler means comprises global scheduler means comprising means for dynamically scheduling and means for communicating said prioritized schedule to said at least one local scheduler.

7. The apparatus of Claim 6 wherein said local scheduler is adapted to communicate information about said plurality of local processes to said global scheduler.

8. The apparatus of Claim 6 wherein said global scheduler further comprises timer means associated with said communication means to periodically effect communication of said dynamically created prioritized schedule to said local schedulers.

9. The apparatus of Claim 6 wherein said global scheduler includes at least one table comprising the identity and address for each of said at least one local scheduler.

10. The apparatus of Claim 2 wherein said scheduler means comprises global scheduler means comprising means for dynamically scheduling and means for communicating said prioritized schedule to said at least one local scheduler.

11. A method for scheduling a plurality of tasks of at least one application among processes on at least one computing node, in a system having scheduler means and at least one computing node, each computing node having a plurality of local processes comprising the steps of:

providing application information to scheduler means;

dynamically creating a prioritized schedule of said plurality of tasks;

determining correspondence between said plurality of tasks and said plurality of local processes; and

dynamically prioritizing said local processes in accordance with said prioritized schedule.

12. The method of Claim 11 wherein said dynamically prioritizing comprises invoking operating system priorities to schedule tasks in accordance with said prioritized schedule.

13. The method of Claim 11 wherein said scheduler means is remotely located from said at least one computing node, further comprising the steps of communicating said prioritized schedule of tasks to said at least one computing node.

14. The method of Claim 12 further comprising the step of said local processes executing said tasks in parallel in accordance with said dynamic prioritizing.

15. The method of Claim 14 further comprising the step of communicating information about execution of said tasks to said remotely located scheduler.

16. The method of Claim 15 further comprising the steps of repeating said steps of dynamically creating a prioritized schedule of said plurality of tasks; determining correspondence between said plurality of tasks and said plurality of local processes; and dynamically prioritizing said local processes in accordance with said prioritized schedule; executing; and communicating information about execution until all tasks have been completed.

17. The method of Claim 14 further comprising the step of interleaving local operations with said executing.

18. The method of Claim 13 further comprising said remotely located scheduler dynamically maintaining at least one list of said at least one computing node.

SYSTEM AND METHOD FOR MAXIMIZING USAGE OF COMPUTER RESOURCES IN SCHEDULING OF APPLICATIONS

ABSTRACT OF THE INVENTION

A task schedule is enforced among multiple processes by setting process priorities based upon which tasks are running on which processes and based upon the task schedule. The task scheduling may be provided by a local or global scheduler which uses application information to prioritize tasks. The task schedule, or priority list, is provided at Local Activity Schedulers which schedule the activities for their local execution elements/nodes. Execution of activities locally are performed by any number of processes that reside in each execution element. These processes are assigned operating system priorities by the respective Local Activity Scheduler based on their assigned activities for execution and the task schedule.

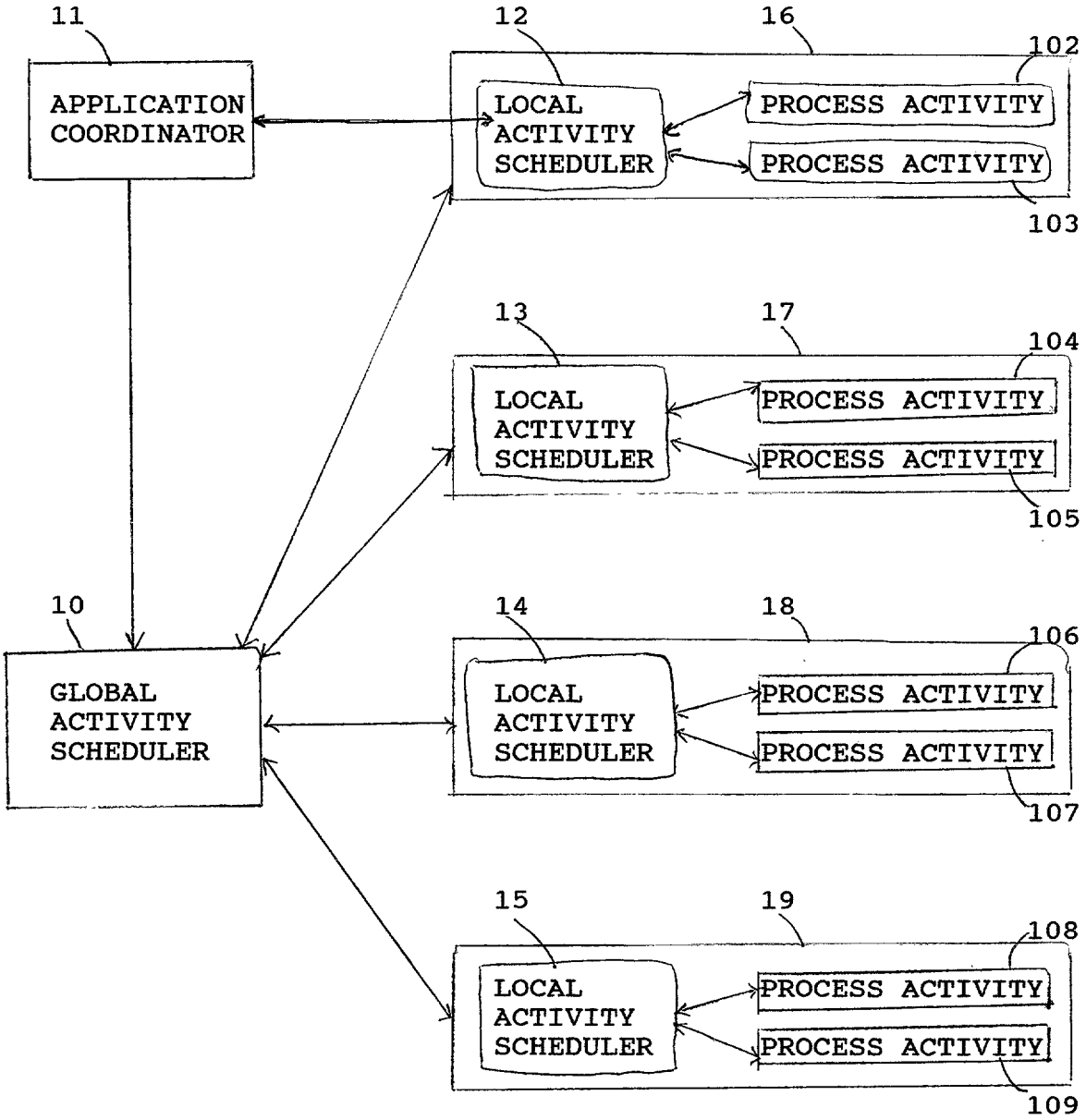
11
APPLICATION COORDINATOR
12
16
102
LOCAL ACTIVITY SCHEDULER
PROCESS ACTIVITY
PROCESS ACTIVITY
103
13
17
104
LOCAL ACTIVITY SCHEDULER
PROCESS ACTIVITY
PROCESS ACTIVITY
105
10
GLOBAL ACTIVITY SCHEDULER
14
18
106
LOCAL ACTIVITY SCHEDULER
PROCESS ACTIVITY
PROCESS ACTIVITY
107
15
19
108
LOCAL ACTIVITY SCHEDULER
PROCESS ACTIVITY
PROCESS ACTIVITY
109

FIGURE 1

ACTIVITY ID
A
B
C
D
⋮

FIGURE 2

LOCAL SCHEDULER ID	LOCAL SCHEDULER LOCATION
0	W
1	X
2	Y
3	Z
⋮	⋮

FIGURE 3

ACTIVITY ID	PROCESS ID	PRIORITY ASSIGNMENT
A	27	1
E	28	2
D	29	3
B	30	4
⋮	⋮	⋮

FIGURE 4

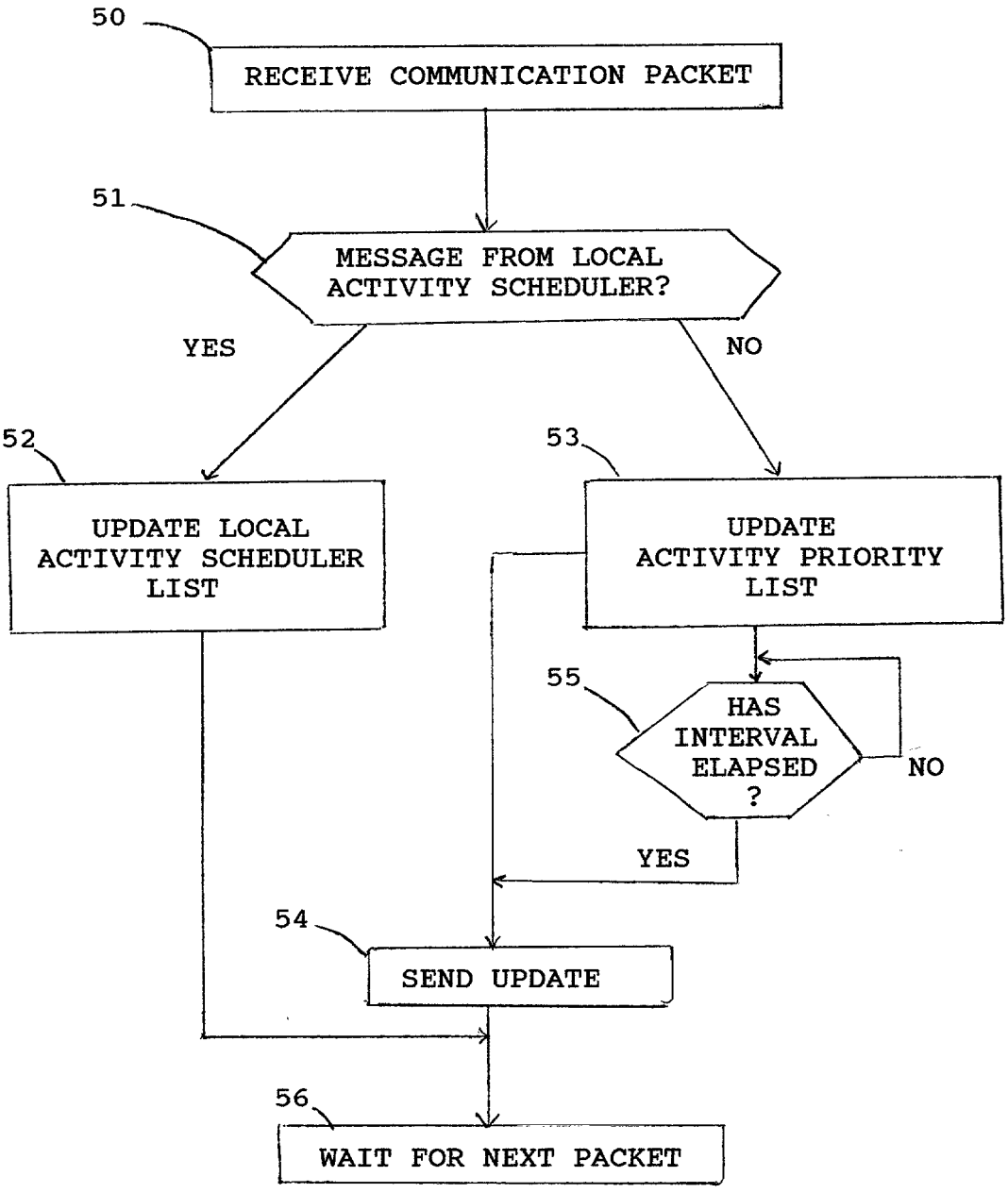
50
RECEIVE COMMUNICATION PACKET
51
MESSAGE FROM LOCAL ACTIVITY SCHEDULER?
YES
NO
52
UPDATE LOCAL ACTIVITY SCHEDULER LIST
53
UPDATE ACTIVITY PRIORITY LIST
55
HAS INTERVAL ELAPSED ?
NO
YES
54
SEND UPDATE
56
WAIT FOR NEXT PACKET

FIGURE 5

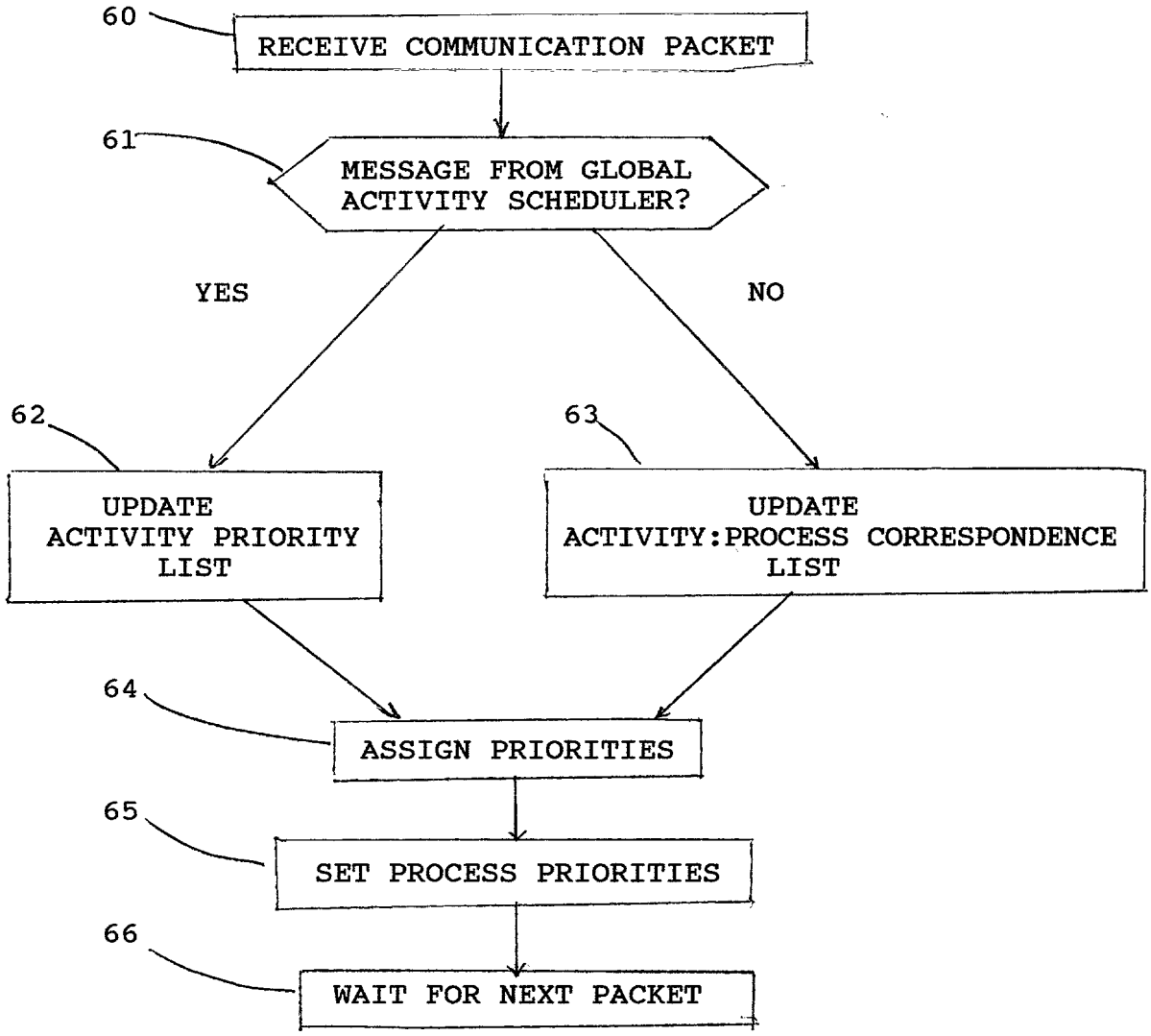
60
RECEIVE COMMUNICATION PACKET
61
MESSAGE FROM GLOBAL ACTIVITY SCHEDULER?
YES
NO
62
UPDATE ACTIVITY PRIORITY LIST
63
UPDATE ACTIVITY:PROCESS CORRESPONDENCE LIST
64
ASSIGN PRIORITIES
65
SET PROCESS PRIORITIES
66
WAIT FOR NEXT PACKET

FIGURE 6

PARALLEL MACHINE

0	B	A	C
1	A	B	C
2	A	C	B
3	C	B	A
TIME	0	10	20	30

FIGURE 7

PRIOR ART

PARALLEL MACHINE

0	B	A	C
1	B	A	C
2	B	A	C
3	B	A	C
TIME	0	10	20	30

FIGURE 8

Docket No. YO997-111

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

SYSTEM AND METHOD FOR MAXIMIZING USAGE OF COMPUTER RESOURCES IN SCHEDULING OF APPLICATION TASKS

the specification of which (check one)

x is attached hereto.

___ was filed on _____ as United States Application Number

or PCT International Application Number _____

and was amended on _____ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or §365(b) of any foreign application(s) for patent or inventor's certificate, or §365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application, having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	___ Yes ___ No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	___ Yes ___ No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	___ Yes ___ No

I hereby claim the benefit under 35 U.S.C. §119(e) of any United States provisional application(s) listed below.

_____ (Application Number)	_____ (Filing Date)
_____ (Application Number)	_____ (Filing Date)

I hereby claim the benefit under 35 U.S.C. §120 of any United States Application(s), or §365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States, or PCT International application in the manner provided by the first paragraph of 35 U.S.C. §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in 37 CFR §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

_____ (Application Serial No.)	_____ (Filing Date)	_____ (Status) (patented, pending, abandoned)
_____ (Application Serial No.)	_____ (Filing Date)	_____ (Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith (list name and registration number).

Manny W. Schecter (Reg. 31,722), Marc D. Schechter (Reg. 28,989), Christopher A. Hughes (Reg. 26,914), Edward A. Pennington (Reg. 32,588), John E. Hoel (Reg. 26,279), Joseph C. Redmond, Jr. (Reg. 18,753), and Douglas W. Cameron (Reg. 31,596).

Send Correspondence to: Douglas W. Cameron, IBM Corporation, P. O. Box 218, Yorktown Heights, NY 10598

Direct Telephone Calls to: (name and telephone number) Douglas W. Cameron (914) 945-3244

Mitchell Adam Cohen
Full name of sole or first inventor

Mitchell Adam Cohe — June 20, 1997
Inventor's Signature — Date

104 Kensington Way, Mt.Kisco, New York 10549
Residence

USA
Citizenship

Same as above.
Post Office Address

Docket No. YO997-111

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

Anant Deep Jhingran
Full name of second joint-inventor, if any

Inventor's signature — Date 6/20/97

47 Nob Hill Drive, Elmsford, New York 10523
Residence

India
Citizenship

Same as above.
Post Office Address

Ronald Mraz
Full name of third joint inventor, if any

Inventor's Signature — Date 6/20/97

169 Smith Ridge Road, South Salem, New York 10590
Residence

USA
Citizenship

Same as above.
Post Office Address

Full name of fourth joint-inventor, if any

Inventor's signature — Date

Residence

Citizenship

Post Office Address

Full name of fifth joint inventor, if any

Inventor's Signature — Date

Residence

Citizenship

Post Office Address

Full name of sixth joint-inventor, if any

Inventor's signature — Date

Residence

Citizenship

Post Office Address

Full name of seventh joint-inventor, if any

Inventor's signature — Date

Residence

Citizenship

Post Office Address